LOAN DOCUMENT

DTIC ACCESSION NUMBER

LEVEL

INVENTORY

DOCUMENT IDENTIFICATION

DISTRIBUTION STATEMENT

H A N D L E W I T H C A R E

ACCESSION FOR

NTIS	GRA&I
DTIC	TRAC
UNANNOUNCED	
JUSTIFICATION	

BY

DISTRIBUTION/

AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL
A-1	

DISTRIBUTION STAMP

DATE ACCESSIONED

DATE RETURNED

19981223 076

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM 70A
JUN 90

DOCUMENT PROCESSING SHEET

LOAN DOCUMENT

NADC

Tech. Info

APPENDIX 19
RP-16 LINKING LOADER
FINAL SOFTWARE REPORT
DATA ITEM NO. A005

INTEGRATED ELECTRONIC WARFARE SYSTEM
ADVANCED DEVELOPMENT MODEL (ADM)

PREPARED FOR:

NAVAL AIR DEVELOPMENT CENTER
WARMINSTER, PENNSYLVANIA

CONTRACT N62269-75-C-0070

RAYTHEON

ELECTROMAGNETIC
SYSTEMS DIVISION

1 OCTOBER 1977

APPENDIX 19

RP-16 LINKING LOADER DEVELOPMENT SPECIFICATION
FINAL SOFTWARE REPORT
DATA ITEM A005

INTEGRATED ELECTRONIC WARFARE SYSTEM (IEWS)
ADVANCED DEVELOPMENT MODEL (ADM)
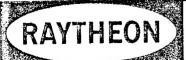
Contract No. N62269-75-C-0070

Prepared for:

Naval Air Development Center
Warminister, Pennsylvania

Prepared by:

RAYTHEON COMPANY
Electromagnetic Systems Division
6380 Hollister Avenue
Goleta, California 93017

1 OCTOBER 1977

TYPE OF SPEC

Computer Program Development Specification

TITLE OF SPEC

RP-16 Linking Loader

FUNCTION	APPROVED	DATE	FUNCTION	APPROVED	DATE
WRITER	N. A. Fujiyoshi	8/6/75			

REVISIONS

CHK	DESCRIPTION	REV	CHK	DESCRIPTION	REV

REVISION																								
SHEET NO.																								
REV STATUS OF SHEETS	REVISION																							
	SHEET NO.																							

1.0 SCOPE

1.1 IDENTIFICATION

The Linking Loader is one of three software units that comprise the RP-16 microprocessor's basic relocatable assembler/loader. The overall software package has been assigned Raytheon control number   TBD   . This document is assigned as a designator CG- TBD   consistent with Technical Standards 3 and 3050. The authorized abbreviation for this loader package is Raytheon Macro Assembler Link Loader or RAMA Link Loader.

1.2 FUNCTIONAL SUMMARY

The Linking Loader Specification is one of three development specifications. The first specification, Raytheon Macro Assembler for the RP-16, CG- TBD   , defines the organization of the load module text for either absolute or relocatable code.

This document specifies the functional characteristics of the RAMA Linking Loader. The Linking Loader is a program which inputs load modules in RAMA Relocatable Object Text format and stores them in RP-16 memory. It is capable of loading modules starting at a location specified by the operator, linking modules, and outputting a load map.

2.0 APPLICABLE DOCUMENTS

The following documents, of the exact issue shown, form a part of this specification to the extent specified herein:

RP-16 Microprocessor Manual (blue book)
Raytheon Macro Assembler for the RP-16 Functional Specification, CG- TBD
RAMA *Relocatable object Text Spec*

3.0 REQUIREMENTS

3.1 SYSTEM PERFORMANCE

The RAMA Linking Loader shall be basically a set of subroutines with distinct entry points that are basic to the overall load/link/map process of a loader for relocatable code. These individual subroutines shall be controlled by a basic operating system to load object text. The loader shall be written such that it will accommodate a variety of load module source devices such as:

High Speed Paper Tape (IOC)   Magnetic Tape
Teletype (IOC)   Cassette Tape
Floppy Disk   Punched Cards
Standard Disk

Subroutines for I/O control of those devices noted as IOC (Initial Operational Capability) shall be accomplished as part of the basic loader.

3.2 ENVIRONMENT

3.2.1 EQUIPMENT CONFIGURATION

The loader shall operate on a system consisting as a minimum of:

a) RP-16 Microprocessor
b) 2K of Memory
c) Teletype or Equivalent
d) High Speed Paper Tape Reader

3.2.2 SOFTWARE CONFIGURATION

The linking leader subroutines shall operate in conjunction with a basic operating system. This basic operating system shall be controlled by operator entry of commands via teletype (or equivalent) and in turn shall call the required subroutines of the Linking Loader. The teletype controlled operating system shall be written as part of the IOC software package for the hardware configuration of 3.2.1.

3.2.3 INTERFACES

All access to peripheral devices shall be by way of I/O subroutines. Provisions shall be made for a simple addition of I/O device drive routines into the Linking Loader package.

3.3 FUNCTIONAL REQUIREMENTS

The RP-16 Linking Loader is a program which inputs load modules in RP-16 Relocatable Object Text format and stores them in memory. It is capable of loading modules starting at a location specified by the operator, linking modules, and outputing a load map.

The Loader is written in RP-16 assembly language and is loaded by bootstrapping it into memory either with the PIN card or the absolute binary loader. It is coded such that it can be loaded and executed starting at any memory location. It will run in an RP-16 with either double-word or byte instruction sets.

The basic loader is written as a set of subroutines with distinct entry points for performing each of five functions. All I/O operations are performed by external calls to an I/O interface routine.

3.3.1 LOADER COMMANDS

The loader command functions are Set Bias Location, Initialize and Load, Load and Link, Output Map, and Start program. The entry points corresponding to these functions are respectively: L.BIAS, L.LOAD, L.LINK, L.MAP, and L.GO.
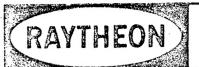
3.3.1.1 Set Bias Location

A subroutine call to L.BIAS causes the loader to set the bias location for loading the next program to the value in the A-register. The calling sequence is

```
        JSUB   (=L.BIAS)
        JUMP   error        error return location
          -                 normal return location
```

If the A-register contains an address taken up by the loader, an error return will be made.

RAYTHEON

RAYTHEON COMPANY
LEXINGTON, MASS. 02173

CODE IDENT NO.
49956

SPEC NO.

SHEET
4 OF 5

REV

3.3.1.2 Initialize and Load

A subroutine call to L.LOAD causes the loader to clear its symbol table and load one program module. The calling sequence is:

```
JSUB    (=L.LOAD)
JUMP    error        .   error return location
  -                      normal return
```

If a loading error is encountered, the loader will perform an error return with the error code in the A-register. The error codes and their meanings are:

1   checksum error
2   sequence number error
3   illegal item type
4   attempt to load over the loader or symbol table
5   load error

3.3.1.3 Load and Link

A subroutine call to L.LINK causes the loader to load one program module without initializing the symbol table. The calling sequence is:

```
JSUB    (=L.LINK)
JUMP    error        error return location
  -                   normal return location
```

If a loading error is encountered, the loader will perform an error return with the error code in the A-register. The error codes are the same as listed in Section 3.3.1.2.

3.3.1.4 Output Map

A subroutine call to L.MAP causes the loader to output the symbol table. The calling sequence is:

```
JSUB    (=L.MAP)
```

The symbol table is first sorted by symbol name, then output with a separate line per symbol name and its value. Undefined symbol names are output with a preceding asterisk and a value equal to the last reference location. The last line of the map contains the next available load location.
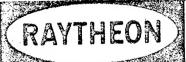
3.3.1.5 Start Program

A subroutine call to L.GO causes the loader to perform a jump to the address contained in the A-register if it is non-zero. Otherwise, it will jump to the location specified by the last item type ØE encountered in the object text. The calling sequence is:

```
JSUB    (=L.GO)
```

3.3.1.6 Loader Operation

The loader builds its symbol table starting with the memory location immediately preceding its first instruction and expanding it toward lower memory. Each entry point or undefined external reference causes a new entry to be added to the symbol table. The maximum amount of object text that can be loaded depends on the amount of memory available and the number of external names used.

The loader performs I/O operations by doing subroutine calls to an I/O interface package. Object text is read by executing the following calling sequence: *with the buffer address in the A-register*

```
        JSUB    (=I.BINI)
        DATA    1
        -                       return location
```

The first word in the buffer contains the value 80_{10} followed by 40 words. The map is output by performing the following calling sequence for each line output: *with the buffer address in the A-register*

```
        JSUB    (=I.COUT)
        DATA    2
        -                       return location
```

The first word of the buffer contains the count of the number of bytes to be output followed by the text in 8-bit ASCII characters.

4.0   QUALITY ASSURANCE

      N/A

5.0   PREPARATION FOR DELIVERY

      N/A

APPENDIX I

OBJECT CODE ITEM TYPES

00 - Null Item (1 byte)

01 - Entry Point Name (9 bytes)
    Name - 8 bytes left justified

02 - Absolute Program Origin (3 bytes)
    Location - 2 bytes

03 - Relocatable Program Origin (3 bytes)
    Location - 2 bytes

04 - Fill Data (5 bytes)
    Word Count - 2 bytes
    Data - 2 bytes

05 - 16-bit Absolute Data (3 bytes)
    Data - 2 bytes

06 - 16-bit Relocatable Data (3 bytes)
    Data - 2 bytes

07 - Absolute Location (3 bytes)
    Location - 2 bytes

08 - Relocatable Location (3 bytes)
    Location - 2 bytes

09 - Entry Point Definition (9 bytes)
    Name - 8 bytes

0A - External Link Pointer (11 bytes)
    Name - 8 bytes
    Offset- 2 bytes

0B - Internal Link Pointer 16-bits (3 bytes)
    Offset - 2 bytes

0C - Internal Link Pointer 8-bits (3 bytes)
    Offset - 2 bytes

0D - Variable Length Data (2 x Count + 1 bytes)
    Word Count - 1 byte
    Data - 2 x Count bytes

0E - Transfer Address (1 byte)

0F - Program Name (9 bytes)
    Name - 8 bytes

FF - End of Load Module (1 byte)